CE349: THEORY OF COMPUTATION

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	-	-	3	**3**
Marks	100	-	-	100	

Pre-requisite courses:

- Mathematics, Data Structures and Algorithms, Design and Analysis of Algorithm (Computational Complexity)

Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1	Introduction	03
2	Mathematical Terms and Theory	05
3	Regular Grammar & Languages, Regular Expression, Finite Automata	14
4	Context Free Grammar & Languages, Push down	13
5	Turing Machine, Recursively Enumerable Languages	08
6	Decidable & Undecidable Problems	02
	Total hours (Theory) :	45
	Total hours (Lab) :	00
	Total hours :	45

Detailed Syllabus:

1.	**Introduction**	**03 Hours**	**06 %**
	Alphabet, String, Language, Formal Grammar, Chomsky Hierarchy, Introduction to Automata		
2.	**Mathematical Terms and Theory**	**05 Hours**	**10 %**
	Mathematical Inductions, Recursive Definitions		
3.	**Regular Grammar & Languages, Regular Expression,**	**14 Hours**	**32 %**

	Finite Automata		
	Regular Language, Regular Expressions, Applications, Chomsky Hierarchy, Finite Automata, Nondeterministic Finite Automata, Kleen's Theorem, Automata with Output (Moore Machine, Mealy Machine), Properties of Regular Languages (Pumping Lemma, Closure Property, Decision Algorithm)		
4.	**Context Free Grammar & Languages, Push down Automata**	**13 Hours**	**30%**
	The Chomsky, Notion of Grammars and Languages Generated by Grammars, CFG, CFL, Regular Language and Regular Grammar, Derivation Tree and Ambiguity, BNF, CNF, GNF, CFL properties (Pumping Lemma, Closure Property, Decision Algorithm), Intersections and Complements of CFL, Non-CFL, Definition, DPDA, NPDA, Equivalence of CFG and PDA		
5.	**Turing Machine, Recursively Enumerable Languages**	**08 Hours**	**18%**
	Definition, Model of Computation, Combining TM, Variations of TM, Non Deterministic TM, Universal TM, Recursively Enumerable and Recursive, Enumerable Languages, Context sensitive languages		
6.	**Decidable & Undecidable Problems**	**02 Hours**	**04 %**
	Tractable and Intractable Problems, Complexity Classes, Tractable and Possibly Intractable Problems, P and NP Completeness, Countability		

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Apply basic concepts of theory of computation in the computer field in order to solve computational problems.
CO2	Construct algorithms for different problems and argue formally about correctness on different restricted machine models of computation.

CE362: MACHINE LEARNING

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	-	5	4
Marks	100	50	-	150	

Pre-requisite courses:

- Probability, linear algebra, calculus and programming language

Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Data cleaning and Preprocessing	06
2.	Supervised Machine Learning	14
3.	Unsupervised Machine Learning	11
4.	Ensemble learning	04
5.	Deep Neural Networks	10
	Total hours (Theory) :	45
	Total hours (Lab) :	30
	Total hours :	75

Detailed Syllabus:

1.	**Data Cleaning and Pre-processing**	**06 Hours**	**13%**
	Handling Missing data, Data Exploration and Visualization, Handling Outliers, Data Integration and Aggregation, Feature selection: Filter Methods, Wrapper Methods, Embedded Methods, Data Transformation: Feature scaling and normalization, Encoding categorical variables: one-hot encoding, label encoding, Data transformation for skewed distributions		
2.	**Supervised Machine Learning**	**14 Hours**	**31%**
	What is supervised ML, Linear Regression for univariate and Multivariate data, Cost function, Gradient Descent, Logistic regression, Under fitting and Over fitting, Support Vector		

	Machine, Decision Tree, Random Forest, Artificial Neural Network architecture, Activation functions, Forward pass in ANN, Back propogation in ANN, Model Evaluation techniques.		
3.	**Unsupervised Machine Learning**	**11 Hours**	**25%**
	What is unsupervised ML, Distance Techniques: Euclidean Distance, Manhattan Distance, Minkowski Distance, Cosine Similarity, Hamming Distance. Different clustering algorithm: K-means, Hierarchical Clustering, DBSCAN, Evaluate clustering algorithm, Evaluation Techniques: Evaluate Clustering metrics - Silhouette Score, Calinski-Harabasz Index, Davies-Bouldin Index, Dunn Index. Dimensionality reduction: Principal Component Analysis(PCA)		
5.	**Ensemble learning**	**04 Hours**	**09%**
	Gradient Boosting, XGBoost (Extreme Gradient Boosting), AdaBoost (Adaptive Boosting)		
5.	**Deep Neural Networks**	**10 Hours**	**22%**
	Introduction to Convolutional Neural Networks, Convolutional Neural Network Layers, CNN Architectures and Variants, Introduction to Recurrent Neural: Understanding sequential data and its challenges, Anatomy of a Recurrent Neural Network (RNN), vanishing gradient problem Networks, Long Short-Term Memory (LSTM) Networks.		

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Addressing data quality issues, effectively cleaning and preprocessing raw datasets, and preparing them for machine learning algorithm.
CO2	Capable of building predictive models for classification and regression tasks, and skilled in selecting, training, and evaluating machine learning algorithms on labeled data

CE361: OPERATING SYSTEM

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	3	2	0	5	**4**
Marks	100	50	0	150	

Pre-requisite courses:

- Introduction to computer and computer organization.

Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction	02
2.	Process Management	06
3.	Process Synchronization	06
4.	Deadlocks	04
5.	Memory Management	10
6.	Secondary Storage and Input Output Management	04
7.	File Systems	04
8.	System Protection	03
9.	Unix/Linux File System	04
10.	Epilogue	02
	Total hours (Theory) :	45
	Total hours (Lab) :	30
	Total hours :	75

Detailed Syllabus:

1.	**Introduction**	**02 Hours**	**05%**
	What is an OS? OS structure, System Calls, Functions of OS, Evolution Of OS, OS design consideration for Multiprocessor and Multicore		
2.	**Process Management**	**06 Hours**	**13%**
	Process: What is Process? Process Control Block, Process States, Process API: fork(), exec() and wait() system calls, Process Scheduling, Threads: Types of Threads, Multithreading model, Scheduler activation		

3.	**Process Synchronization**	**06 Hours**	**13%**
	Race Conditions, Critical Section, Peterson's Solution, hardware primitives for synchronization, Mutex Locks, Semaphores, Monitors and Classical problems in synchronization(Producer-Consumer, Reader's & Writer Problem, Dinning Philosopher Problem).		
4.	**Deadlocks**	**04 Hours**	**9%**
	Modelling, Characterization, Prevention and Avoidance, Detection and Recovery		
5.	**Memory Management**	**10 Hours**	**22%**
	Address Space, Address Translation, Multiprogramming with Fixed and variable partitions, Swapping, Segmentation, Free Space Management, Paging: Principle Of Operation, Page Allocation, H/W Support For Paging. Multilevel paging, Virtual Memory: Concept, Performance of Demand Paging, Page Replacement Algorithms, Thrashing and Working Sets.		
6.	**Secondary Storage and Input Output Management**	**04 Hours**	**09%**
	I/O Devices: Device Controllers, Device Drivers, Direct Memory Access. Goals of The I/O S/W, Interrupt Handler, Device Independent I/O Software Disks: Disk architecture, Disks Scheduling Algorithm, RAID		
7.	**File Systems**	**04 Hours**	**09%**
	File: File Naming, File attributes, File Operations, File Structure, File Types, Access Methods, Directories: Structure, Pathnames, Directory Operations. File System Implementation: partitioning and mounting, Allocation methods: Contiguous Allocation, Linked Allocation, Indexed Allocation.		
8.	**System Protection**	**03 Hours**	**07%**
	Goals of protection, Principles of protection, Domain of protection, Access matrix, Implementation of Access matrix, Access control		
9.	**Unix/Linux File System**	**04 Hours**	**09%**
	Buffer Cache, Inodes, The system calls - ialloc, ifree, namei, alloc and free. Mounting and Unmounting , files systems, Network File systems. EXT file system in linux		
10.	**Epilogue**	**02 hours**	**04%**
	Pointers to advanced topics: distributed OS, multimedia OS, embedded OS, real-time OS, OS for multiprocessor machines, research problems in OS		

CE385: Mobile Application Development

Credits and Hours:

Teaching Scheme	Theory	Practical	Tutorial	Total	Credit
Hours/week	2	2	-	4	**3**
Marks	100	50	-	150	

Outline of the Course:

Sr. No.	Title of the unit	Minimum number of hours
1.	Introduction to Kotlin	02
2.	Functional Programming	03
3.	Objected oriented programming	05
4.	Dart Programming	06
5.	Flutter User Interface (Flutter UI)	06
6.	Theming, Styling and Routing	08
	Total hours (Theory) :	30
	Total hours (Lab) :	30
	Total hours :	60

Detailed Syllabus:

1.	**Introduction to Kotlin**	**02 Hours**	**6%**
	Introduction, Why Kotlin? Functions, Looping & Ranges, Objects Everywhere, Creating Class, Constructors, Packages, Exceptions, Enumerations, Data Classes, Safe Calls & the Elvis Operator, Introduction to Generics.		
2.	**Functional Programming**	**03 Hours**	**10%**
	Lambdas, Higher-Order Functions, Lists, Maps, Recursion		
3.	**Objected oriented programming**	**05 Hours**	**17%**

	Interfaces, Secondary Constructors, Inheritance, Abstract Classes, Upcasting, Polymorphism, Composition, Inheritance & Extensions, Class Delegation, Down casting,Nested Classes, Objects, Inner Classes, Companion Objects		
4.	**Dart Programming**	**06 Hours**	**20%**
	Dart Introduction, Stagehand –project generator, The pubspec file, Introducing async programming with features		
5.	**Flutter User Interface (Flutter UI)**	**06 Hours**	**20%**
	Introduction to Flutter, widgets introduction, hello flutter, Understanding built-in widgets, Creating UI with widgets, creating custom widgets		
6.	**Theming, Styling and Routing**	**08 Hours**	**27%**
	Themes widgets, Using custom fonts, Dynamic styling with MediaQuery, Understanding the navigator widget. Names routes, Screen transitions, Hero animations.		

Course Outcome (COs):

At the end of the course, the students will be able to

CO1	Knowledge of Kotlin programming language: Solid understanding of the Kotlin programming language.
CO2	Understanding of flutter framework: Through understanding of the flutter framework and its architecture.
CO3	Proficiency in developing mobile application: Developing mobile applications using the flutter and Kotlin
CO4	Ethical consideration: Understand the ethical considerations involved in mobile application development

Course Articulation Matrix:

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	1	2	2	1	2	-	-	-	-	-	-	-	1	-
CO2	3	2	3	1	3	-	-	-	-	-	-	-	2	-